

Итоговая аттестация по курсу Java.

Модуль: Клиент-серверные приложения на Java (JDBC)

Тема: PostgreSQL, JDBC, Maven, работа с БД из Java

Цель работы

Разработать и развернуть приложение для учёта заказов, используя PostgreSQL.

Приложение должно включать:

1. Создание и миграцию схемы БД (DDL + тестовые данные).
2. Набор SQL-запросов на чтение/изменение/удаление.
3. Java-приложение на Maven, которое подключается к этой БД через JDBC и выполняет базовые операции CRUD, выводя результаты в консоль.

Задача 1. Установка и настройка PostgreSQL

1. Рекомендована 12 версия PostgreSQL. Установить PostgreSQL на локальную машину <https://postgrespro.ru/docs/postgrespro/14/binary-installation-on-windows> или использовать Docker-контейнер.
2. Проверить подключение к PostgreSQL из IDE IntelliJ Idea <https://www.jetbrains.com/help/idea/postgresql.html#connect-to-postgresql-database>

Упрощение задачи 1. Реализовать схему в онлайн-редакторе. Например, в редакторе <https://sqliteonline.com/>. Обратите внимание, что реализовать требуется с помощью PostgreSQL.

Задача 2. Проектирование и создание схемы БД

1. **Файл `schema.sql`** должен содержать:
 - Таблицы:
 - **product** — id, описание, стоимость, количество, категория.
 - **customer** — id, имя, фамилия, телефон, email.
 - **order** — id, product_id (FK), customer_id (FK), дата заказа, количество, статус.
 - **order_status** — справочник статусов заказов (id, имя статуса).
 - Первичные и внешние ключи, ограничения NOT NULL, CHECK (например, цена ≥ 0 , количество ≥ 0).
 - Индексы по внешним ключам и дате заказа.
 - Комментарии к таблицам и ключевым полям.
 - CREATE TABLE IF NOT EXISTS, чтобы скрипт можно было запускать повторно.
2. **Заполнение тестовыми данными** — минимум 10 строк в каждой таблице.

* Схема данных может быть выбрана любой на усмотрение студента. Можно проявить фантазию и создать Базу Данных на ваш лад.

Задача 3. SQL-запросы

Создать файл **test-queries.sql**, включающий не менее 10 запросов:

- 5 запросов на **чтение** (в том числе JOIN с несколькими таблицами, агрегаты, сортировка, фильтрация).
- 3 запроса на **изменение** (UPDATE).
- 2 запроса на **удаление** (DELETE).

Примеры:

- Список всех заказов за последние 7 дней с именем покупателя и описанием товара.
- Топ-3 самых популярных товара.
- Обновление количества на складе при покупке.
- Удаление клиентов без заказов.

Задача 4. Java-приложение (Maven, без Spring)

1. Создать Maven-проект (Java 17) с зависимостями:
 - PostgreSQL JDBC Driver
 - (опционально) Flyway для миграций
2. Реализовать класс `App.java`, который:
 - Подключается к БД через JDBC (`DriverManager`).
 - Выполняет автоматический запуск миграций (если используется Flyway).
 - Демонстрирует CRUD-операции:
 1. Вставка нового товара и покупателя (`PreparedStatement`).
 2. Создание заказа для покупателя.
 3. Чтение и вывод последних 5 заказов с JOIN на товары и покупателей.
 4. Обновление цены товара и количества на складе.
 5. Удаление тестовых записей.
 - Работает в транзакции с `commit()` и `rollback()` при ошибках.
 - Выводит результаты операций в консоль в удобочитаемом формате.
3. Параметры подключения (URL, user, password) вынести в `application.properties`

Результат

1. Репозиторий на GitHub (ветка + Pull Request).
2. Скриншоты:
 - Установка/подключение PostgreSQL.
 - Выполнение миграций.
 - ER-диаграмма или схема БД.
 - Запуск Java-приложения и вывод CRUD-операций.
 - Результаты выполнения `test-queries.sql` в IDE.

